Instituto Tecnológico de Costa Rica

Área Académica de Ingeniería en Computadores

Curso: CE-3101 Bases de Datos

Proyecto II - StraviaTEC

Profesor: Marco Rivera Meneses

Estudiantes:

Gabriel Vargas López

Víctor Castrillo Muñoz

Yendry Badilla Gutiérrez

Mauricio Calderón Chavarría

Grupo: 1

Fecha: 6 de junio, 2022

Tabla de Contenidos

Modelo conceptual

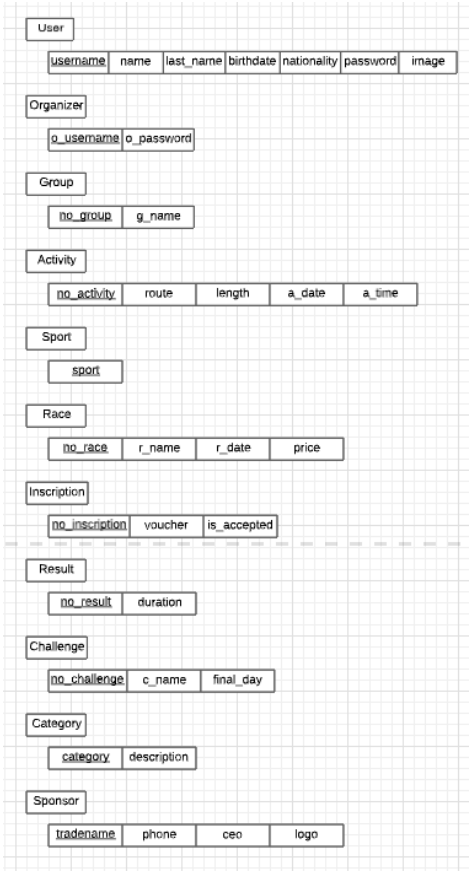
El modelo conceptual se adjunta al final del documento y en PDF en el repositorio, para mayor calidad de imagen.

Modelo relacional y descripción detallada del mapeo

Tanto el modelo relacional así como el conceptual se pueden visualizar de mejor manera en el siguiente enlace: https://lucid.app/lucidchart/7a73255f-df78-4f7f-bccd-1a88a8636bc2/edit?viewport_loc=1082%2C-1652%2C7172%2C3540%2C0_0&invitationId=inv_b91d565f-2e5e-4bb4-bd60-cce0925f1767#

Primeramente, antes de realizar el mapeo, se decidió pasar las entidades al modelo correcto antes de comenzar con las relaciones, esto para tener orden durante el procedimiento.

Para el mapeo como primer paso se identificaron las entidades fuertes y se realizó el mapeo de estas escribiendo los atributos simples y compuestos en las tablas, identificando con subrayado las primary keys de cada una de las entidades.



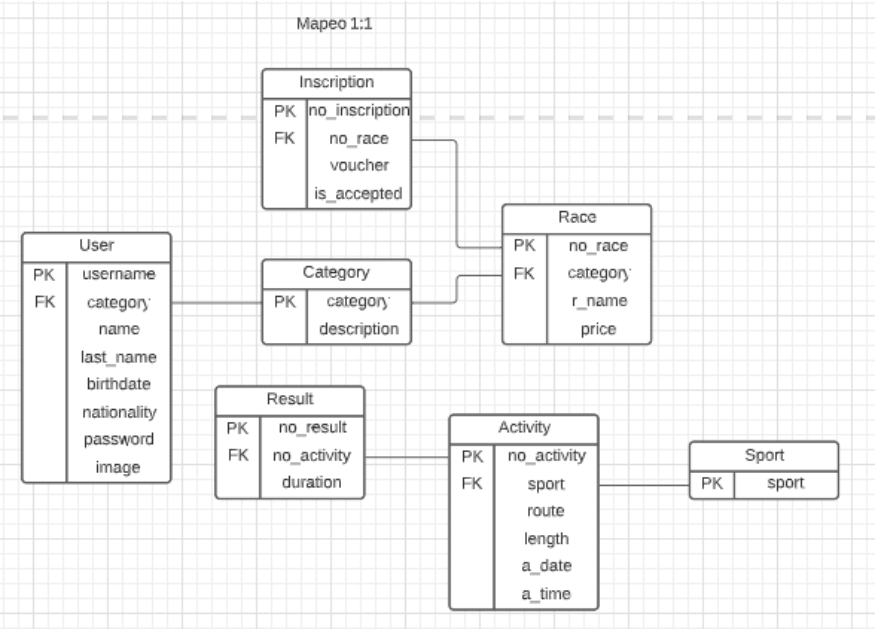
Luego debido a que nuestro caso en especifico no presenta entidades débiles, procedemos a realizar el mapeo 1 a 1, en ese mismo momento se van agregando e identificando las llaves foráneas, tal y como se muestra a continuación.

En nuestro caso las entidades 1:1 a destacar son

Inscription →Race Category → Race

Result → Activity User → Category

Activity → Sport

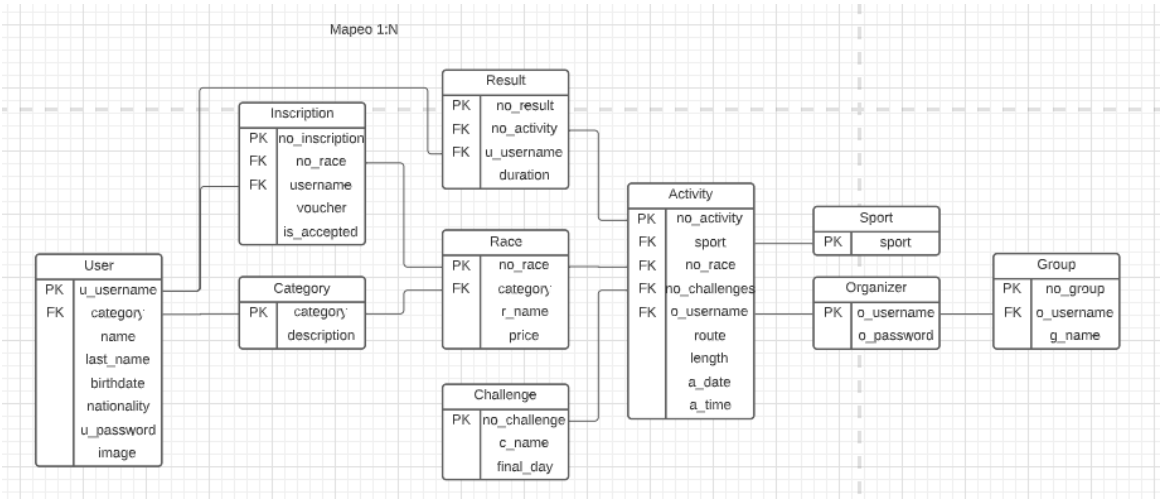


Posteriormente pasamos al mapeo de entidades 1 a N, en este ya podemos ver como nos va quedando mas grande el modelo, es importante recalcar la importancia marcas las llaves foráneas respectivas a la hora de hacer el mapeo. Las relaciones identificadas 1:N son:

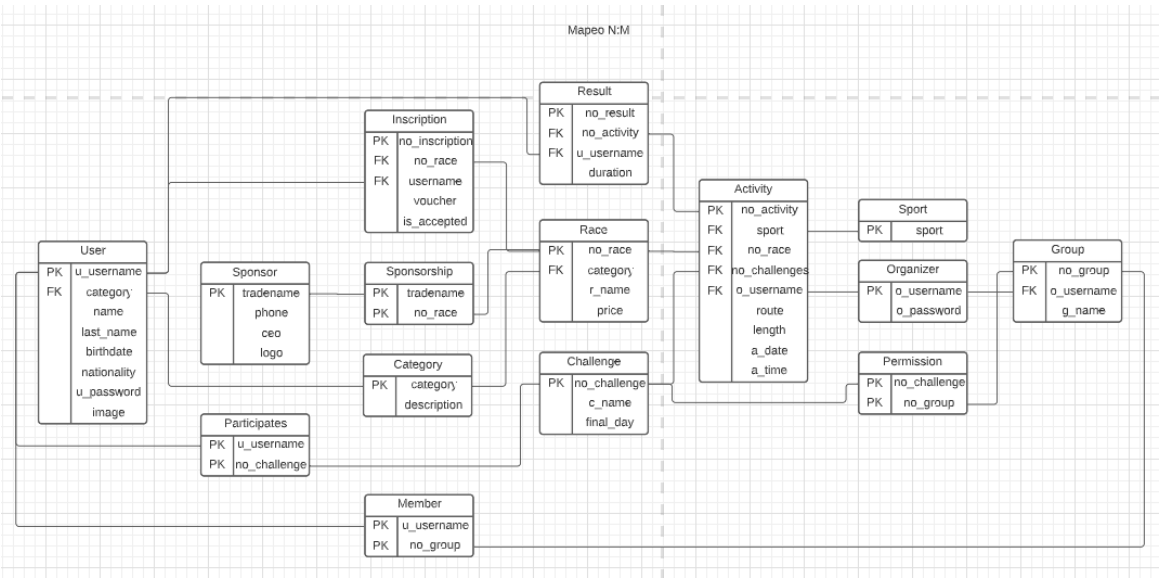
Inscription → User Group → Organizer

Challenge→ Activity Organizer → Activity

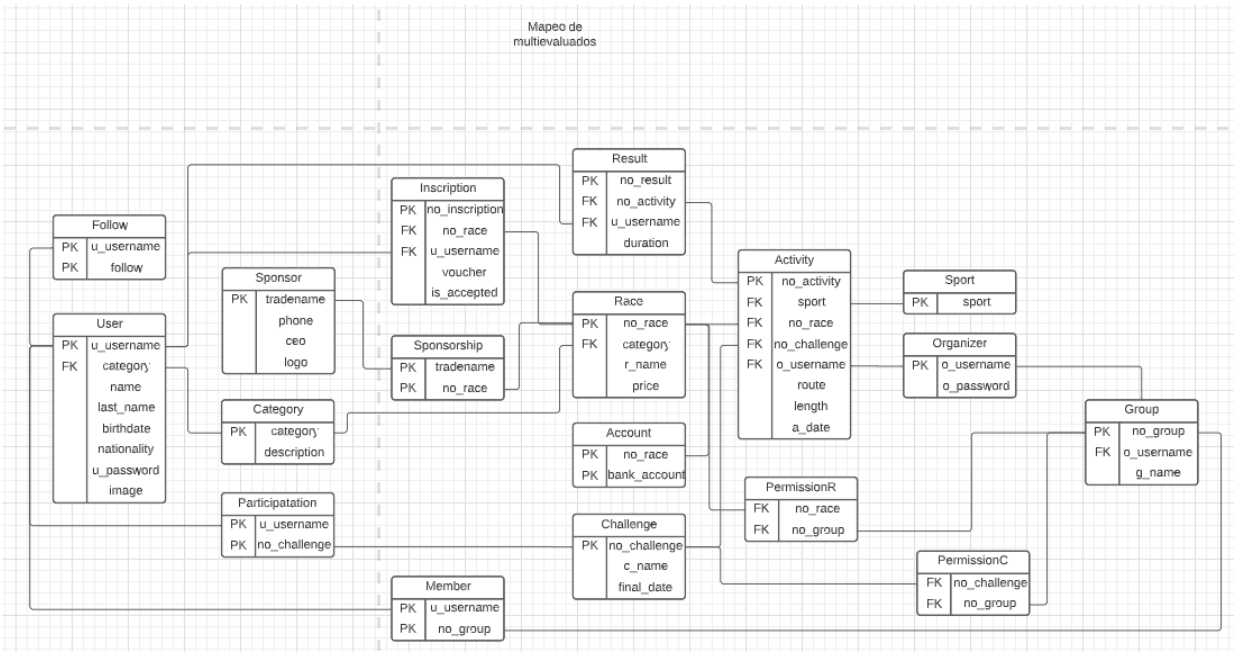
Result → User Inscription → Race



Luego pasamos al mapeo de N:M, para este debemos tener en cuenta que las relaciones se llevan a cabo a través de tablas intermedias que almacenan las dos llaves primarias de las relaciones que se están uniendo, tal es el caso de member, que almacena la llave foránea de gruop y de user entre otros. El modelo después de aplicar este paso de mapeo se observa a continuación



Y por último para terminar el mapeo es necesario agregar los multivaluados, en este caso se debe crear una tabla que relacione a este atributo con su entidad, tal es el caso de bank_accounts o follower. Luego de realizar esto ya tendríamos completado todos los pasos de mapeo y tendríamos nuestro modelo conceptual final, tal y como se muestra a continuación



Si se desea ver el mapeo en una imagen de mas calidad, es mejor acceder al siguiente enlace:

https://lucid.app/lucidchart/7a73255f-df78-4f7f-bccd-1a88a8636bc2/edit?viewport_loc=1082%2C-1652%2C7172%2C3540%2C0_0&invitationId=inv_b91d565f-2e5e-4bb4-bd60-cce0925f1767#

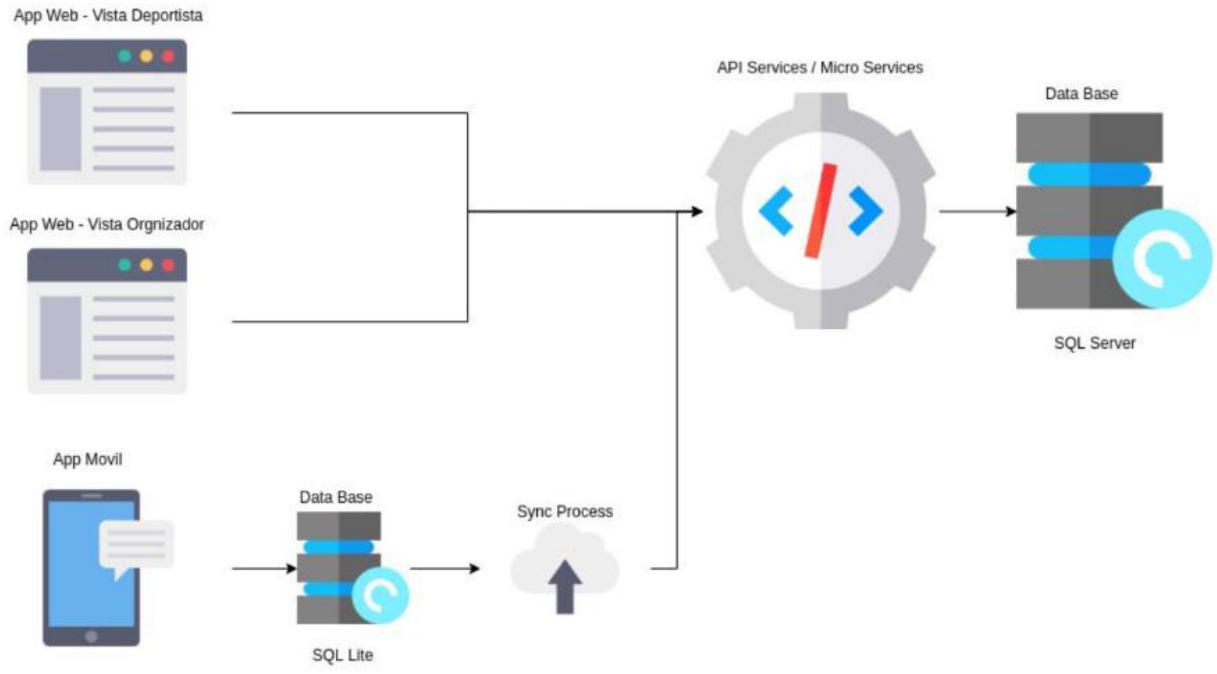
Descripción de las estructuras de datos desarrolladas (Tablas)

La tabla User contiene todos los atributos que permiten realizar la autenticación de cuentas, como el usuario, nombre, apellido, fecha de nacimiento, nacionalidad y contraseña. El usuario contiene una llave foránea de la categoría, la cual es una tabla que solo contiene la descripción de esta. Estas categorías se utilizan para clacificar las carreras, las cuales contienen su nombre y precio.

La actividad es una tabla más general que contiene llaves foráneas de deporte, carrera, reto y usuario, además de atributos propios como la ruta que realiza, su longitud en kilómetros y la fecha en la que se realiza. El reto es una tabla que contiene su nombre y fecha a realizar. Este tiene anclado una tabla resultado que contiene un usuario y una duración, y una tabla del organizador que puede crear actividades, retos y carreras que solo tiene un atributo con el usuario y la contraseña para acceder a esta cuenta.

Descripción detallada de la arquitectura desarrollada

Según la descripción del caso expuesto el equipo decidió optar por la siguiente arquitectura:



Para este caso en específico se nos pidió desplegar dos API, el equipo en cuestión solo desplego un API, en donde se puede acceder a las dos bases de datos utilizadas (Mongo DB y SQL Server) ambas desplegadas en la Azure.

EL API también se desplego en Azure y este se desarrolló en .NET Framework y por último, las vistas web se ubicaron en una sola web page desarrollada en Angular y desplegada también en Azure

Conexiones Web →API: Para realizar la conexión Web con API se utilizó el protocolo Https, en este caso en especial al estar el API alojado en Azure, se debe cambiar la url que se utiliza normalmente para realizar diferentes procedimientos como el Post, Get, Put etc. A través de esta conexión es que se pasa toda la información que el usuario brinda y también se recibe toda la información proveniente de la base de datos.

Conexiones API →SQL Server: Para esta conexión se utilizo el connection string que nos desplegaba el Azure al alojar en el la base de datos, esto nos permitió almacenar la totalidad de nuestros datos en la nube.

Conexiones API →MongoDB: Al igual que con la conexión de SQL Server, la base de datos en MongoDB fue desplegada en Azure y con el connection string respectivo se logró hacer la conexión para guardar los comentarios

Conexiones App Móvil →API: Para la app móvil se utilizó una biblioteca llamada Retrofit que a través de los services muy semejantes a los manejados en la parte web, se lograran crear enlaces con el API y por ende con la Base de Datos, para luego cierta información almacenarla en la base local de SQL Lite

Según los requerimientos del proyecto, la base de datos de SQL Server desplegada en Azure posee ya distintos valores por defecto para ciertas tablas según lo mencionado en el anunciado del proyecto. Los valores por defecto de las tablas se muestran a continuación.

Default Activities:

Tipos de Actividad
Running
Cycling
Hiking
Kayaking
Swimming
Walking

Default Categories:

Categoria	Descripción
Junior	menor de 15 años
Sub-23	de 15 a 23 años
Open	de 24 a 30 años
Elite	cualquiera que quiera inscribirse.
Master A	30 a 40 años,
Master B	de 41 a 50 años,
Master C	más de 51 años

Default Sponsors:

Patrocinador	CEO	Numero de contacto	Logo
Adidas	Kasper Rørsted	22016275	
Coca-Cola	James Quincey	22472800	
Crystal	Darner Mora Alvarado	84962514	
Garmin	Clifton A. Pemble	934972373	
Kolbi	Charlie Mora	20025146	
Nike	Mark Parker	22216106	
Nuun	David Mutzel	83810665	
Powerade	Gary Smith	22169827	
Speedo	Andrew Rubin	22016321	

Problemas conocidos

Error en comentarios MongoDB

En la página web se decidió omitir los comentarios debido a que no se logró implementar correctamente, sin embargo, la base de datos en MongoDB se encuentra activa la funcionalidad se podría agregar en cualquier momento.

Error en inicio de mapa en Android Studio

Al iniciar la actividad que muestra el mapa, este aparece en África por unos segundos hasta que el GPS detecta movimiento o se presiona el botón de ubicación actual (esquina superior derecha).

Error al generar el archivo GPX en Android Studio

Se intentó implementar la biblioteca JPX para crear el GPX, pero por la forma en la que se almacenan las latitudes y longitudes en el MapActivity no se pudo utilizar la clase GPX que es la que se encargaba de crear el GPX. Debido a esto se investigó sobre otra forma de generar el archivo, la cual escribía un string con formato de XML y GPX, utilizando la lista de coordenadas generadas por la ruta, posteriormente se traducía a documento XML, se implementó en el código sin embargo al finalizar la actividad, la app se cerraba y se reportaba que el error era que no podía generar el archivo a partir de un NULL, sin embargo, la lista si estaba guardando las coordenadas.

Error al girar el teléfono cuando se muestra el mapa

Cuando se muestra el mapa, este va a tener registrado el movimiento del usuario con una línea en el mapa. Al rotar el teléfono, todo el proceso se elimina, por lo que se recomiendo bloquear la rotación automática al probar la aplicación en un teléfono.

Problemas encontrados

Error al enviar archivos GPX

Durante el desarrollo, a lo hora de adjuntar el gpx, se tuvieron problemas para enviar dicho archivo al API para su posterior almacenaje, esto se debía a la falta de headers en la consulta especifica, esto logro solventar este problema y nos permitió almacenarlos correctamente.

Error al cargar imágenes

En ciertas partes de la web (foto de perfil, váucher de pago) se quería poder subir una imagen con los recursos necesarios, pero no fue posible, por lo que se optó por subir el link al recurso en línea, en vez de adjuntar los archivos locales del sistema.

Error al imprimir los GPX como mapas interactivos

Para poder representar el GPX file como un mapa interactivo se optó inicialmente por utilizar los servicios de Google para ello, sin embargo, se presentaron problemas con el API Key de servicio y finalmente se decidió utilizar un API Key de Open Layers código abierto para poder acceder a los mapas interactivos y así poder interpretar el GPX como tal.

Error en la vista del mapa

En las secciones de perfil y dashboard de la web, no se logró poner la vista del mapa directamente en la tarjeta de resumen de actividad, por el contrario, la solución planteada fue sustituir el mapa por un botón que me permitiera abrir la vista de mapa.

Error al agregar múltiples valores al crear carreras/retos (cuentas bancarias, categorías, sponsors)

Durante el desarrollo del proyecto se nos presentó un reto respecto a cómo manejar los datos múltiples variables a nivel de desarrollo web, debido a que no se presentaba una manera sencilla de poder ofrecer una vista en tiempo real de los datos conforme estos se fueran agregando en la vista de creador, para solucionar esto se implementó un sistema de actualización a través de un request al API, con esto las tablas de vista múltiple quedaron funcionales.

Error en button navigation view

Al implementar el menú de navegación en la aplicación móvil, no se mostraba en la pantalla, para solucionar esto se utilizó el constraint widget para eliminar el espacio vacío que se creaba entre el Fragment y el Button navigation view.

Error al abrir la siguiente actividad

Al realizar la conexión con el API, al cambiar de la actividad de login al menú de seleccionar entrenamientos, la app se cerraba ya que se cambiaba a training fragment, pero se debía cambiar main activity.

Error utilizando API de Google Maps en Web

Para las vistas Web no se logró implementar Google Maps por medio de la API Key, por lo que se decidió utilizar una biblioteca Open Layers que accede a un API open source llamado Openstreet map que brinda los planos para mostrar los mapas en la vista.

Conclusiones

La arquitectura es de un nivel superior, permite resolver el problema planteado y cumplir los objetivos establecidos en el proyecto, implementando servicios de aplicación Web con tecnologías actuales como Angular, conexión por medio de un REST API desarrollado como un WEB API en ASP.net core, bases de datos de gran calidad y fáciles de utilizar por ser open source como SQL Server, desarrollo de aplicaciones móviles en Android Studio, uno de los sistemas operativos más usados en dispositivos móviles.

De acuerdo con lo anterior, cabe mencionar que esta arquitectura y tecnologías son muy utilizadas en la industria de desarrollo de software tanto a nivel nacional como internacional, por lo que se adquirieron habilidades técnicas que se podrán utilizar en el ámbito profesional en el futuro.

Recomendaciones

- Mantener todas las branches del repositorio actualizadas para evitar conflictos de los archivos.
- Evitar subir archivos que generan los IDEs que contienen rutas locales de la computadora de cada desarrollador para evitar conflictos en los archivos.
- Crear las tablas de la base de datos en orden en el script de base de datos puede evitar errores de creación, ya que la base de datos no puede crear primero una tabla que posee una llave foránea a otra tabla que se crea después en el script.

- Es importante consultar la documentación de los lenguajes y tecnologías utilizadas en el desarrollo, ya que entre una versión y otra puede haber cambios en la estructura y forma de estos.

Bibliografía

Android Studio App Development. (2020, 29 octubre). *Fragment Button Click* [Vídeo]. YouTube. https://www.youtube.com/watch?v=x9QB_XXO-uk

ASP.NET Core Dependency Injection error: Unable to resolve service for type while attempting to activate. (2016, 30 noviembre). Stack Overflow. https://stackoverflow.com/questions/40900414/asp-net-core-dependency-injection-error-unable-to-resolve-service-for-type-whil

Calingasan, A. (2021, 13 abril). *.Net 5 Web API With Repository Pattern Docker and PostgreSQL*. Alexcodetuts. https://alexcodetuts.com/2021/04/10/net-5-web-api-with-repository-pattern-docker-and-postgresql/

doctor code. (2020, 2 abril). *Make a Timer App in Android Studio and Java* [Vídeo]. YouTube. https://www.youtube.com/watch?v=4aYhSOyAYQQ

Fonseca, C. (2013, 2 agosto). *List of timed coordinates to a GPX file.* Github. https://gist.github.com/carlosefonseca/6143182

Kamere, R. [larn tech]. (2020, 15 agosto). *Android Login Register with an API: Retrofit Configuration* [Vídeo]. YouTube. https://www.youtube.com/watch?v=LyM5v9WqhTM

Kumar, P. (2020, 10 diciembre). *Java Convert String to XML Document and XML Document to String*. JournalDev. https://www.journaldev.com/1237/java-convert-string-to-xml-document-and-xml-document-to-string

Munonye, K. (2021, 5 junio). *How to Create REST API in .Net Using C# and Visual Studio*. Kindson The Genius. https://www.kindsonthegenius.com/how-to-create-rest-api-in-net-using-c-and-visual-studio/

Munonye, K. [Kindson The Tech Pro]. (2021, 9 junio). *Build your First REST API in Net with C# in Visual Studio - Step by Step* [Vídeo]. YouTube. https://www.youtube.com/watch?v=SDHlLmkdd3s

Nzivu, B. (2021, 5 enero). *Simple GET request using Retrofit in Android*. Section Engineering Education Program. https://www.section.io/engineering-education/making-api-requests-using-retrofit-android/

Sosa, L. (2019, 8 noviembre). *Android Studio: How to draw a real time route and save it?* Stack Overflow. https://stackoverflow.com/questions/58772599/android-studio-how-to-draw-a-real-time-route-and-save-it

Square. (2022a, febrero 27). *Logging Interceptor*. GitHub. https://github.com/square/okhttp/tree/master/okhttp-logging-interceptor

Square. (2022b, abril 28). *Retrofit*. Github. https://square.github.io/retrofit/